IEDB
Immune Epitope Database & Tools

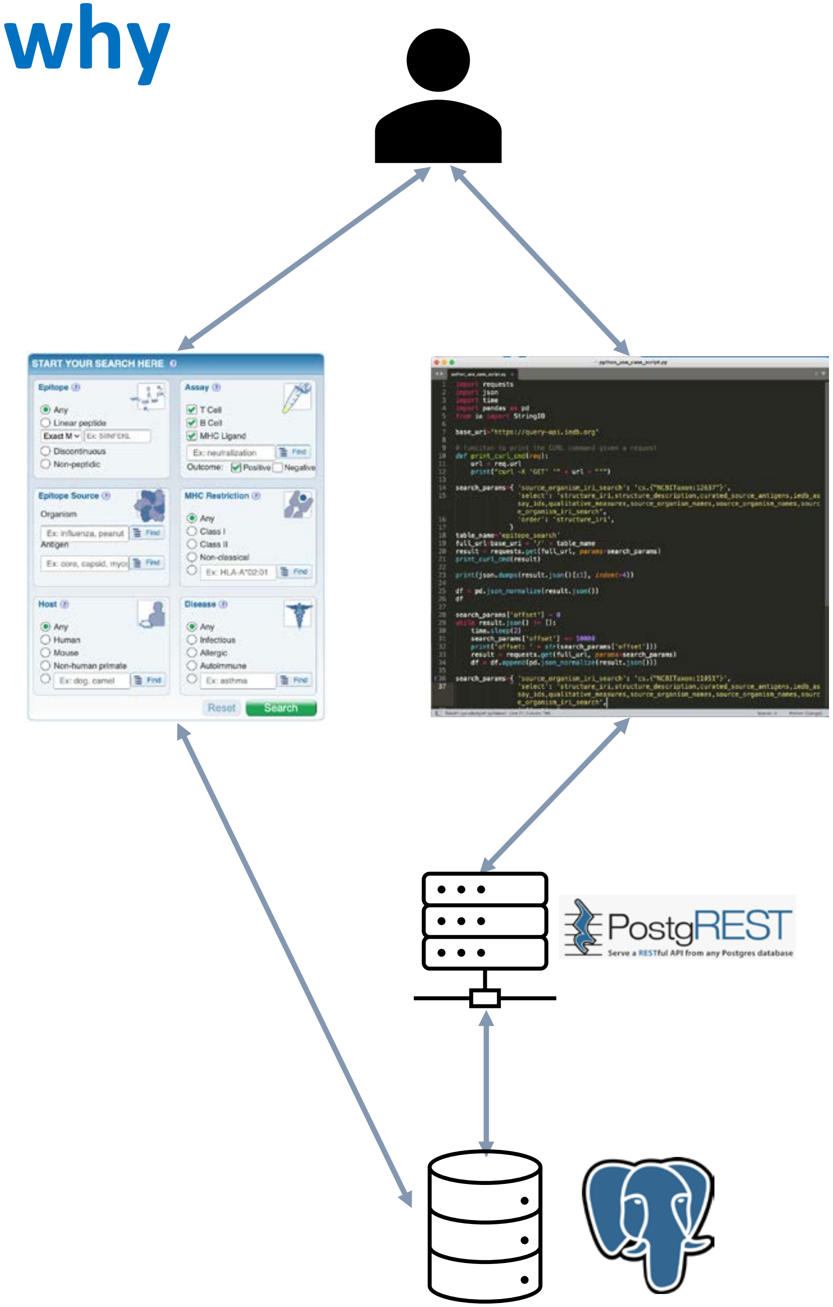
IEDB Query API (IQ-API)

A simple and powerful programmatic query interface to the IEDB

Presented by: J. Greenbaum, Bioinformatics Core Director

What is the IQ-API and why does it exist?

- API = Application Programming Interface
- Alternative interface to query the IEDB
- Content identical to web-based version, but format is different
- Common interface for web server and other applications
- Used for ~850 queries per day



START YOUR SEARCH HERE

Epitope

- ● Any
- ○ Linear peptide
- Exact M ⌄ | Ex: SIINFEKL
- ○ Discontinuous
- ○ Non-peptidic

Assay

- ☑ T Cell
- ☑ B Cell
- ☑ MHC Ligand
- Ex: neutralization | Find
- Outcome: ☑ Positive □ Negative

Epitope Source

- Organism: Ex: influenza, peanut | Find
- Antigen: Ex: core, capsid, myo | Find

MHC Restriction

- ● Any
- ○ Class I
- ○ Class II
- ○ Non-classical
- ○ Ex: HLA-A*02:01 | Find

Host

- ● Any
- ○ Human
- ○ Mouse
- ○ Non-human primate
- ○ Ex: dog, camel | Find

Disease

- ● Any
- ○ Infectious
- ○ Allergic
- ○ Autoimmune
- ○ Ex: asthma | Find

Reset | Search

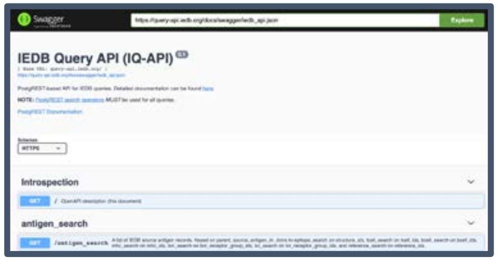
python_use_case_script.py

```python
import requests
import json
import time
import pandas as pd
from io import StringIO

base_url="https://query-api.iedb.org"

# function to print the CURL command given a request
def print_curl_cmd(req):
    url = req.url
    print("curl -X 'GET' '" + url + "'")

search_params={ 'source_organism_iri_search': 'cs.{"NCBITaxon:12637"}',
                'select': 'structure_iri,structure_description,curated_source_antigens,iedb_assay_ids,qualitative_measures,source_organism_names,source_organism_names,source_organism_iri_search',
                'order': 'structure_iri',
              }
table_name='epitope_search'
full_url=base_url + '/' + table_name
result = requests.get(full_url, params=search_params)
print_curl_cmd(result)

print(json.dumps(result.json()[:1], indent=4))

df = pd.json_normalize(result.json())
df

search_params['offset'] = 0
while result.json() != []:
    time.sleep(2)
    search_params['offset'] += 10000
    print('offset: ' + str(search_params['offset']))
    result = requests.get(full_url, params=search_params)
    df = df.append(pd.json_normalize(result.json()))

search_params={ 'source_organism_iri_search': 'cs.{"NCBITaxon:11053"}',
                'select': 'structure_iri,structure_description,curated_source_antigens,iedb_assay_ids,qualitative_measures,source_organism_names,source_organism_names,source_organism_iri_search',
```

How does it work?

API specification available at the base URL:

https://query-api.iedb.org



IEDB Query API (IQ-API)

[Base URL: query-api.iedb.org/]
https://query-api.iedb.org/docs/swagger/iedb_api.json

PostgREST-based API for IEDB queries. Detailed documentation can be found here

NOTE: PostgREST search operators MUST be used for all queries.

PostgREST Documentation

Schemes
HTTPS

Introspection

GET / OpenAPI description (this document)

antigen_search

GET /antigen_search A list of IEDB source antigen records. Keyed on parent_source_antigen_iri. Joins to epitope_search on structure_ids, tcell_search on tcell_ids, bcell_search on bcell_ids, mhc_search on mhc_ids, tcr_search on tcr_receptor_group_ids, bcr_search on bcr_receptor_group_ids, and reference_search on reference_ids.

Each database table is exposed as a separate endpoint, e.g.:

/epitope_search

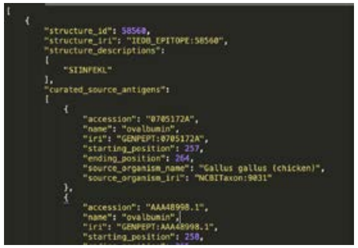
/tcell_search

/antigen_search

…

Queries formulated using PostgREST’s expressive syntax, e.g.,

linear_sequence=eq.SIINFEKL



```
[
    {
        "structure_id": 58560,
        "structure_iri": "IEDB_EPITOPE:58560",
        "structure_descriptions":
        [
            "SIINFEKL"
        ],
        "curated_source_antigens":
        [
            {
                "accession": "0705172A",
                "name": "ovalbumin",
                "iri": "GENPEPT:0705172A",
                "starting_position": 257,
                "ending_position": 264,
                "source_organism_name": "Gallus gallus (chicken)",
                "source_organism_iri": "NCBITaxon:9031"
            },
            {
                "accession": "AAA48998.1",
                "name": "ovalbumin",
                "iri": "GENPEPT:AAA48998.1",
                "starting_position": 258,
```

Putting it all together:

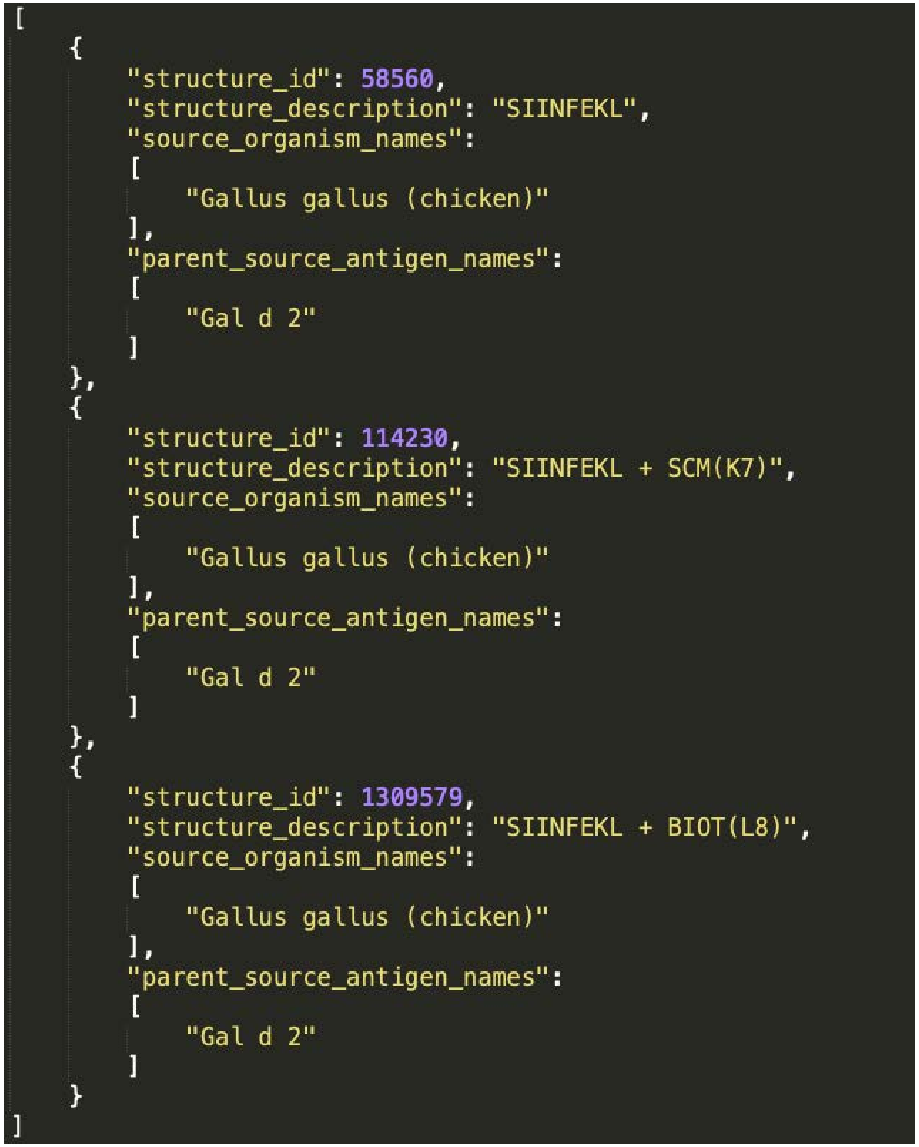
https://query-api.iedb.org/epitope_search?linear_sequence=eq.SIINFEKL

How are results returned?

Epitope query:

https://query-api.iedb.org/epitope_search?linear_sequence=eq.SIINFEKL

Response includes one ‘row’ per epitope:



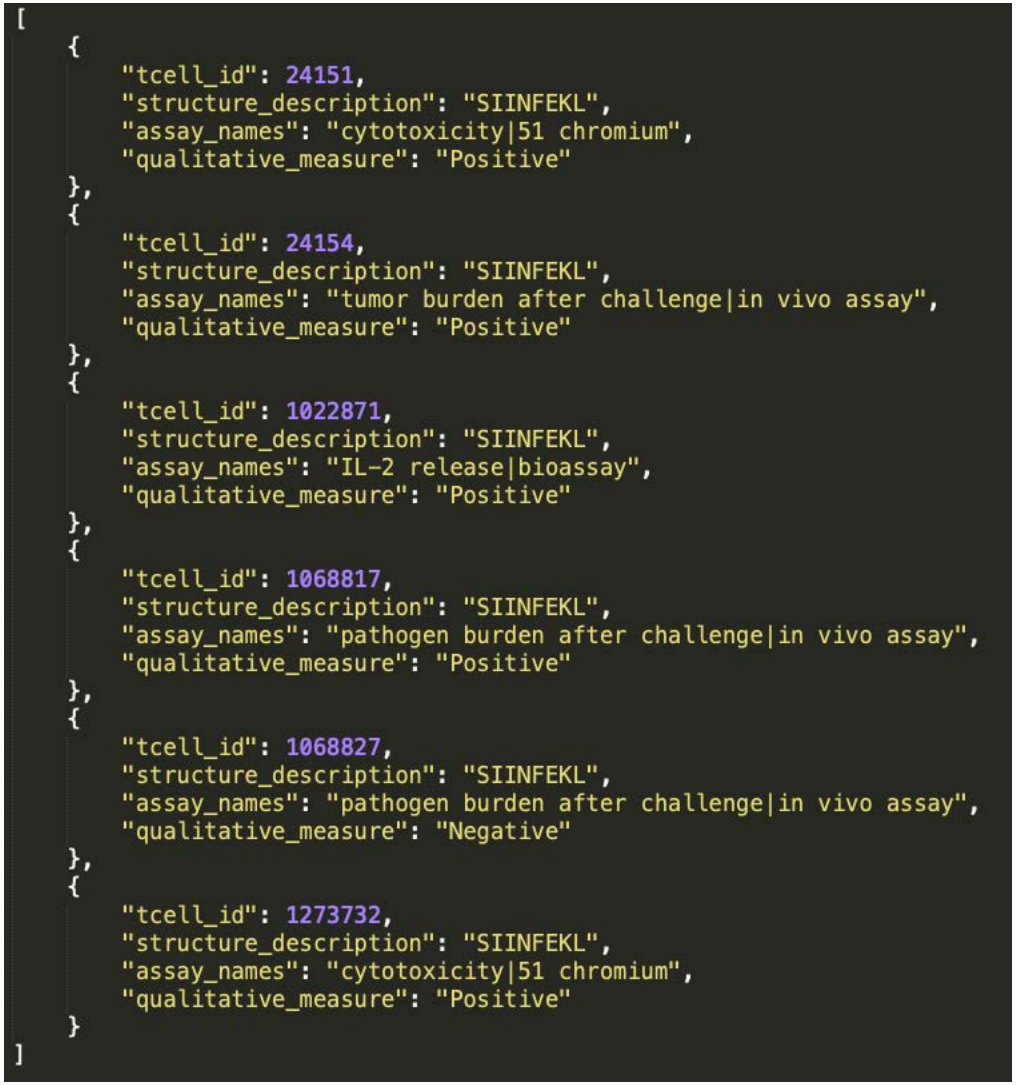
```
[
    {
        "structure_id": 58560,
        "structure_description": "SIINFEKL",
        "source_organism_names":
        [
            "Gallus gallus (chicken)"
        ],
        "parent_source_antigen_names":
        [
            "Gal d 2"
        ]
    },
    {
        "structure_id": 114230,
        "structure_description": "SIINFEKL + SCM(K7)",
        "source_organism_names":
        [
            "Gallus gallus (chicken)"
        ],
        "parent_source_antigen_names":
        [
            "Gal d 2"
        ]
    },
    {
        "structure_id": 1309579,
        "structure_description": "SIINFEKL + BIOT(L8)",
        "source_organism_names":
        [
            "Gallus gallus (chicken)"
        ],
        "parent_source_antigen_names":
        [
            "Gal d 2"
        ]
    }
]
```

How are results returned?

T cell assay query:

https://query-api.iedb.org/tcell_search?linear_sequence=eq.SIINFEKL

Response includes one ‘row’ per T cell assay:

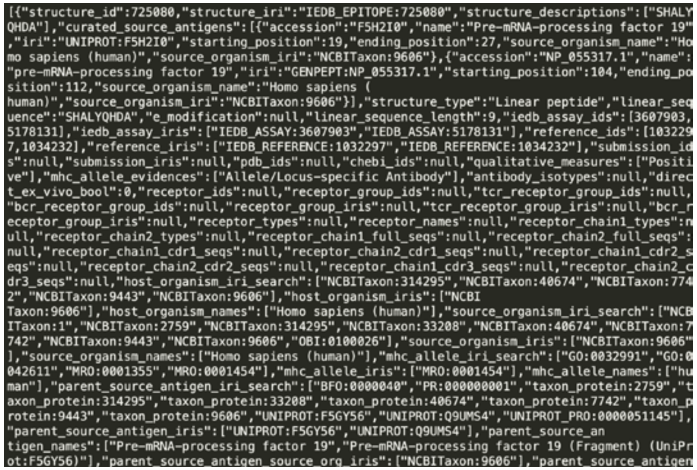


```
[
    {
        "tcell_id": 24151,
        "structure_description": "SIINFEKL",
        "assay_names": "cytotoxicity|51 chromium",
        "qualitative_measure": "Positive"
    },
    {
        "tcell_id": 24154,
        "structure_description": "SIINFEKL",
        "assay_names": "tumor burden after challenge|in vivo assay",
        "qualitative_measure": "Positive"
    },
    {
        "tcell_id": 1022871,
        "structure_description": "SIINFEKL",
        "assay_names": "IL-2 release|bioassay",
        "qualitative_measure": "Positive"
    },
    {
        "tcell_id": 1068817,
        "structure_description": "SIINFEKL",
        "assay_names": "pathogen burden after challenge|in vivo assay",
        "qualitative_measure": "Positive"
    },
    {
        "tcell_id": 1068827,
        "structure_description": "SIINFEKL",
        "assay_names": "pathogen burden after challenge|in vivo assay",
        "qualitative_measure": "Negative"
    },
    {
        "tcell_id": 1273732,
        "structure_description": "SIINFEKL",
        "assay_names": "cytotoxicity|51 chromium",
        "qualitative_measure": "Positive"
    }
]
```

How can I refine my query?

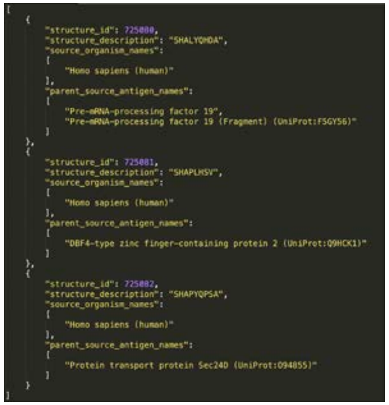
’Horizontal’ filtering: use the ‘select’ keyword to return only the relevant fields

https://query-api.iedb.org/epitope_search



```
[{"structure_id":725080,"structure_iri":"IEDB_EPITOPE:725080","structure_descriptions":["SHALY
QHDA"],"curated_source_antigens":[{"accession":"F5H2I0","name":"Pre-mRNA-processing factor 19"
,"iri":"UNIPROT:F5H2I0","starting_position":19,"ending_position":27,"source_organism_name":"Ho
mo sapiens (human)","source_organism_iri":"NCBITaxon:9606"},{"accession":"NP_055317.1","name":
"pre-mRNA-processing factor 19","iri":"GENPEPT:NP_055317.1","starting_position":104,"ending_po
sition":112,"source_organism_name":"Homo sapiens (
human)","source_organism_iri":"NCBITaxon:9606"}],"structure_type":"Linear peptide","linear_seq
uence":"SHALYQHDA","e_modification":null,"linear_sequence_length":9,"iedb_assay_ids":[3607903,
5178131],"iedb_assay_iris":["IEDB_ASSAY:3607903","IEDB_ASSAY:5178131"],"reference_ids":[103229
7,1034232],"reference_iris":["IEDB_REFERENCE:1032297","IEDB_REFERENCE:1034232"],"submission_id
s":null,"submission_iris":null,"pdb_ids":null,"chebi_ids":null,"qualitative_measures":["Positi
ve"],"mhc_allele_evidences":["Allele/Locus-specific Antibody"],"antibody_isotypes":null,"direc
t_ex_vivo_bool":0,"receptor_ids":null,"receptor_group_ids":null,"tcr_receptor_group_ids":null,
"bcr_receptor_group_ids":null,"receptor_group_iris":null,"tcr_receptor_group_iris":null,"bcr_r
eceptor_group_iris":null,"receptor_types":null,"receptor_names":null,"receptor_chain1_types":n
ull,"receptor_chain2_types":null,"receptor_chain1_full_seqs":null,"receptor_chain2_full_seqs":
null,"receptor_chain1_cdr1_seqs":null,"receptor_chain2_cdr1_seqs":null,"receptor_chain1_cdr2_s
eqs":null,"receptor_chain2_cdr2_seqs":null,"receptor_chain1_cdr3_seqs":null,"receptor_chain2_c
dr3_seqs":null,"host_organism_iri_search":["NCBITaxon:314295","NCBITaxon:40674","NCBITaxon:774
2","NCBITaxon:9443","NCBITaxon:9606"],"host_organism_iris":["NCBI
Taxon:9606"],"host_organism_names":["Homo sapiens (human)"],"source_organism_iri_search":["NCB
ITaxon:1","NCBITaxon:2759","NCBITaxon:314295","NCBITaxon:33208","NCBITaxon:40674","NCBITaxon:7
742","NCBITaxon:9443","NCBITaxon:9606","OBI:0100026"],"source_organism_iris":["NCBITaxon:9606"
],"source_organism_names":["Homo sapiens (human)"],"mhc_allele_iri_search":["GO:0032991","GO:0
042611","MRO:0001355","MRO:0001454"],"mhc_allele_iris":["MRO:0001454"],"mhc_allele_names":["hu
man"],"parent_source_antigen_iri_search":["BFO:0000040","PR:000000001","taxon_protein:2759","t
axon_protein:314295","taxon_protein:33208","taxon_protein:40674","taxon_protein:7742","taxon_p
rotein:9443","taxon_protein:9606","UNIPROT:F5GY56","UNIPROT:Q9UMS4","UNIPROT_PRO:000051145"],
"parent_source_antigen_iris":["UNIPROT:F5GY56","UNIPROT:Q9UMS4"],"parent_source_an
tigen_names":["Pre-mRNA-processing factor 19","Pre-mRNA-processing factor 19 (Fragment) (UniPr
ot:F5GY56)"],"parent_source_antigen_source_org_iris":["NCBITaxon:9606"],"parent_source_antigen
```

https://query-api.iedb.org/epitope_search
?select=structure_id,structure_description,
source_organism_names,
parent_source_antigen_names



```
[
    {
        "structure_id": 725080,
        "structure_description": "SHALYQHDA",
        "source_organism_names":
        [
            "Homo sapiens (human)"
        ],
        "parent_source_antigen_names":
        [
            "Pre-mRNA-processing factor 19",
            "Pre-mRNA-processing factor 19 (Fragment) (UniProt:F5GY56)"
        ]
    },
    {
        "structure_id": 725081,
        "structure_description": "SHAPLHSV",
        "source_organism_names":
        [
            "Homo sapiens (human)"
        ],
        "parent_source_antigen_names":
        [
            "DBF4-type zinc finger-containing protein 2 (UniProt:Q9HCK1)"
        ]
    },
    {
        "structure_id": 725082,
        "structure_description": "SHAPYQPSA",
        "source_organism_names":
        [
            "Homo sapiens (human)"
        ],
        "parent_source_antigen_names":
        [
            "Protein transport protein Sec24D (UniProt:O94855)"
        ]
    }
]
```

‘Vertical’ filtering: query at the field level for specific values

https://query-api.iedb.org/epitope_search

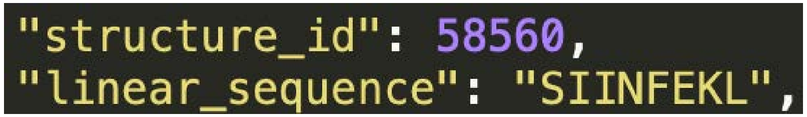
>1,000,000 records returned

https://query-api.iedb.org/epitope_search
?linear_sequence=eq.SIINFEKL

3 records returned

How do I use the PostgREST search operators?

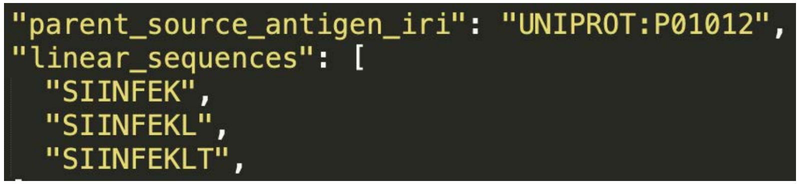
EQUALS (eq.) on string/numeric fields – e.g., linear_sequence in the epitope_search table



```
"structure_id": 58560,
"linear_sequence": "SIINFEKL",
```

linear_sequence=eq.SIINFEKL

CONTAINS (cs.) on list fields – e.g., linear_sequences in the antigen_search table

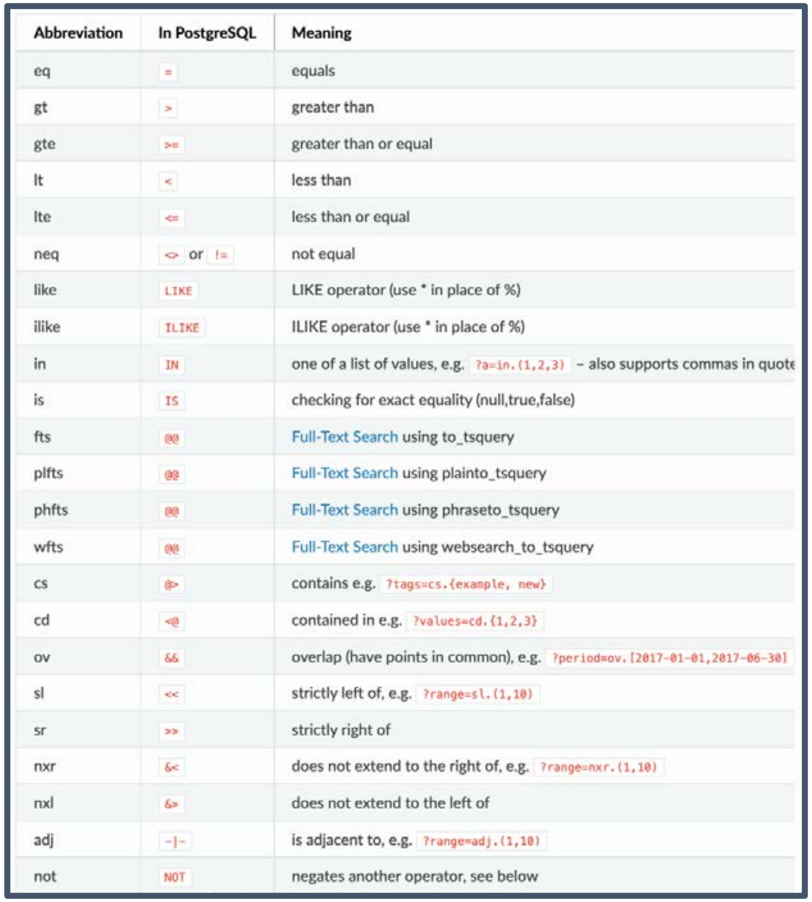


```
"parent_source_antigen_iri": "UNIPROT:P01012",
"linear_sequences": [
  "SIINFEK",
  "SIINFEKL",
  "SIINFEKLT",
```

linear_sequences=cs.{SIINFEKL}

All PostgREST operators

https://postgrest.org/en/stable/api.html#operators



Abbreviation	In PostgreSQL	Meaning
eq	`=`	equals
gt	`>`	greater than
gte	`>=`	greater than or equal
lt	`<`	less than
lte	`<=`	less than or equal
neq	`<>` or `!=`	not equal
like	`LIKE`	LIKE operator (use * in place of %)
ilike	`ILIKE`	ILIKE operator (use * in place of %)
in	`IN`	one of a list of values, e.g. `?a=in.(1,2,3)` – also supports commas in quote
is	`IS`	checking for exact equality (null,true,false)
fts	`@@`	Full-Text Search using to_tsquery
plfts	`@@`	Full-Text Search using plainto_tsquery
phfts	`@@`	Full-Text Search using phraseto_tsquery
wfts	`@@`	Full-Text Search using websearch_to_tsquery
cs	`@>`	contains e.g. `?tags=cs.{example, new}`
cd	`<@`	contained in e.g. `?values=cd.{1,2,3}`
ov	`&&`	overlap (have points in common), e.g. `?period=ov.[2017-01-01,2017-06-30]`
sl	`<<`	strictly left of, e.g. `?range=sl.(1,10)`
sr	`>>`	strictly right of
nxr	`&<`	does not extend to the right of, e.g. `?range=nxr.(1,10)`
nxl	`&>`	does not extend to the left of
adj	`-|-`	is adjacent to, e.g. `?range=adj.(1,10)`
not	`NOT`	negates another operator, see below

Query parameters are similar across all tables

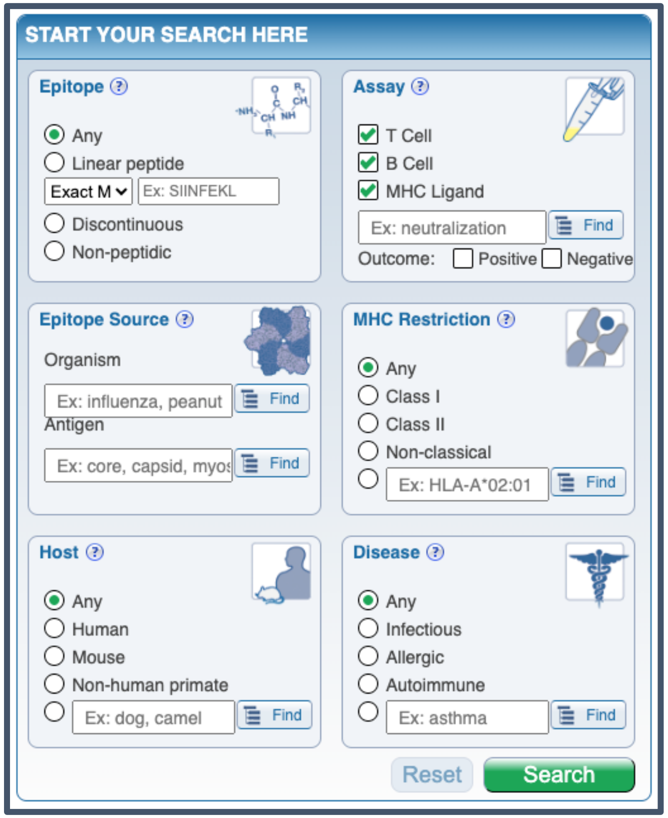
Epitopes:	https://query-api.iedb.org/epitope_search?linear_sequence=eq.SIINFEKL
Antigens:	https://query-api.iedb.org/antigen_search?linear_sequences=cs.{SIINFEKL}
T cell assays:	https://query-api.iedb.org/tcell_search?linear_sequence=eq.SIINFEKL
B cell assays:	https://query-api.iedb.org/bcell_search?linear_sequence=eq.SIINFEKL
MHC assays:	https://query-api.iedb.org/mhc_search?linear_sequence=eq.SIINFEKL
TCRs:	https://query-api.iedb.org/tcr_search?linear_sequences=cs.{SIINFEKL}
BCRs:	https://query-api.iedb.org/bcr_search?linear_sequences=cs.{SIINFEKL}
References:	https://query-api.iedb.org/reference_search?linear_sequences=cs.{SIINFEKL}

Why would I want to use the API?

- Direct access to computer-friendly (JSON) and human-friendly (TSV) formats without web interface
- Incorporate IEDB queries into code
- Work with IEDB data in language of choice
- Easily link between IEDB and other resources
- Periodically rerun automated queries for new/changed data

What can I query?

- If it’s available in the IEDB web interface, it ***should*** be available through IQ-API
- We love a challenge – send us your use cases!



START YOUR SEARCH HERE

Epitope

● Any
○ Linear peptide
Exact M Ex: SIINFEKL
○ Discontinuous
○ Non-peptidic

Assay

☑ T Cell
☑ B Cell
☑ MHC Ligand
Ex: neutralization Find
Outcome: □ Positive □ Negative

Epitope Source

Organism
Ex: influenza, peanut Find
Antigen
Ex: core, capsid, myos Find

MHC Restriction

● Any
○ Class I
○ Class II
○ Non-classical
○ Ex: HLA-A*02:01 Find

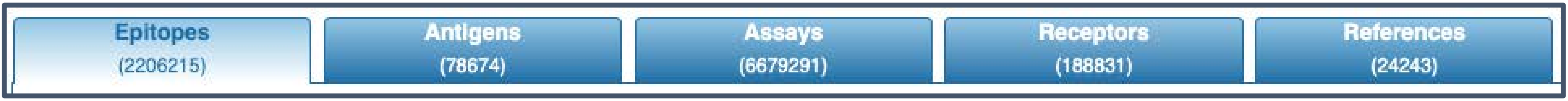
Host

● Any
○ Human
○ Mouse
○ Non-human primate
○ Ex: dog, camel Find

Disease

● Any
○ Infectious
○ Allergic
○ Autoimmune
○ Ex: asthma Find

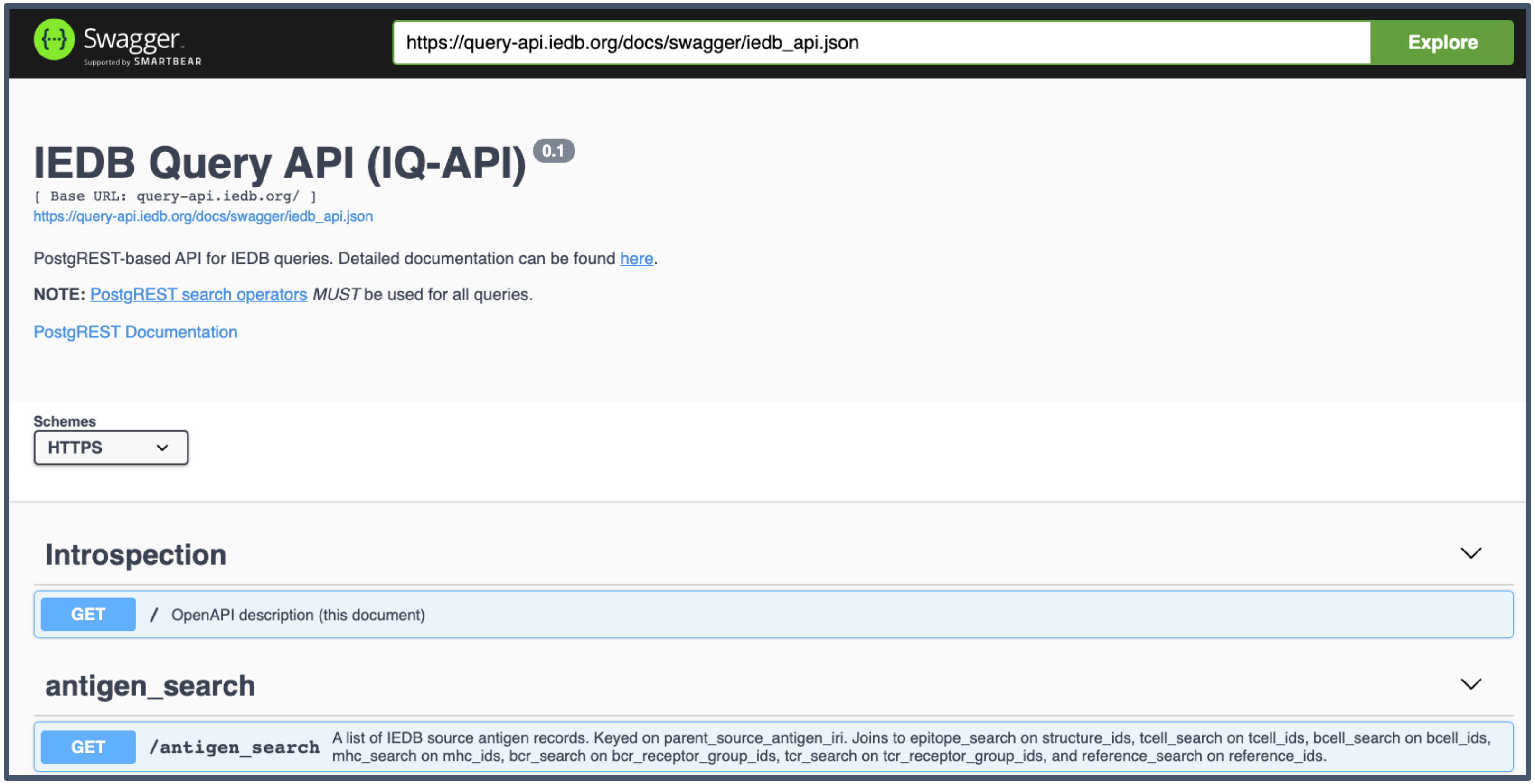
Reset Search



Epitopes	Antigens	Assays	Receptors	References
(2206215)	(78674)	(6679291)	(188831)	(24243)

How can I get started?

https://query-api.iedb.org/docs/swagger/



IEDB Query API (IQ-API) 0.1

[Base URL: query-api.iedb.org/]

https://query-api.iedb.org/docs/swagger/iedb_api.json

PostgREST-based API for IEDB queries. Detailed documentation can be found here.

NOTE: PostgREST search operators *MUST* be used for all queries.

PostgREST Documentation

Schemes

HTTPS

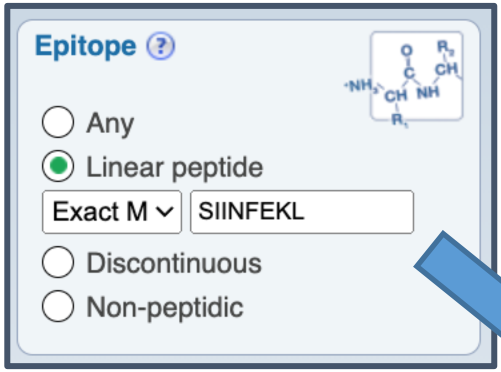
Introspection

GET / OpenAPI description (this document)

antigen_search

GET `/antigen_search` A list of IEDB source antigen records. Keyed on parent_source_antigen_iri. Joins to epitope_search on structure_ids, tcell_search on tcell_ids, bcell_search on bcell_ids, mhc_search on mhc_ids, bcr_search on bcr_receptor_group_ids, tcr_search on tcr_receptor_group_ids, and reference_search on reference_ids.

Reproducing a website query through IQ-API



Epitope ?

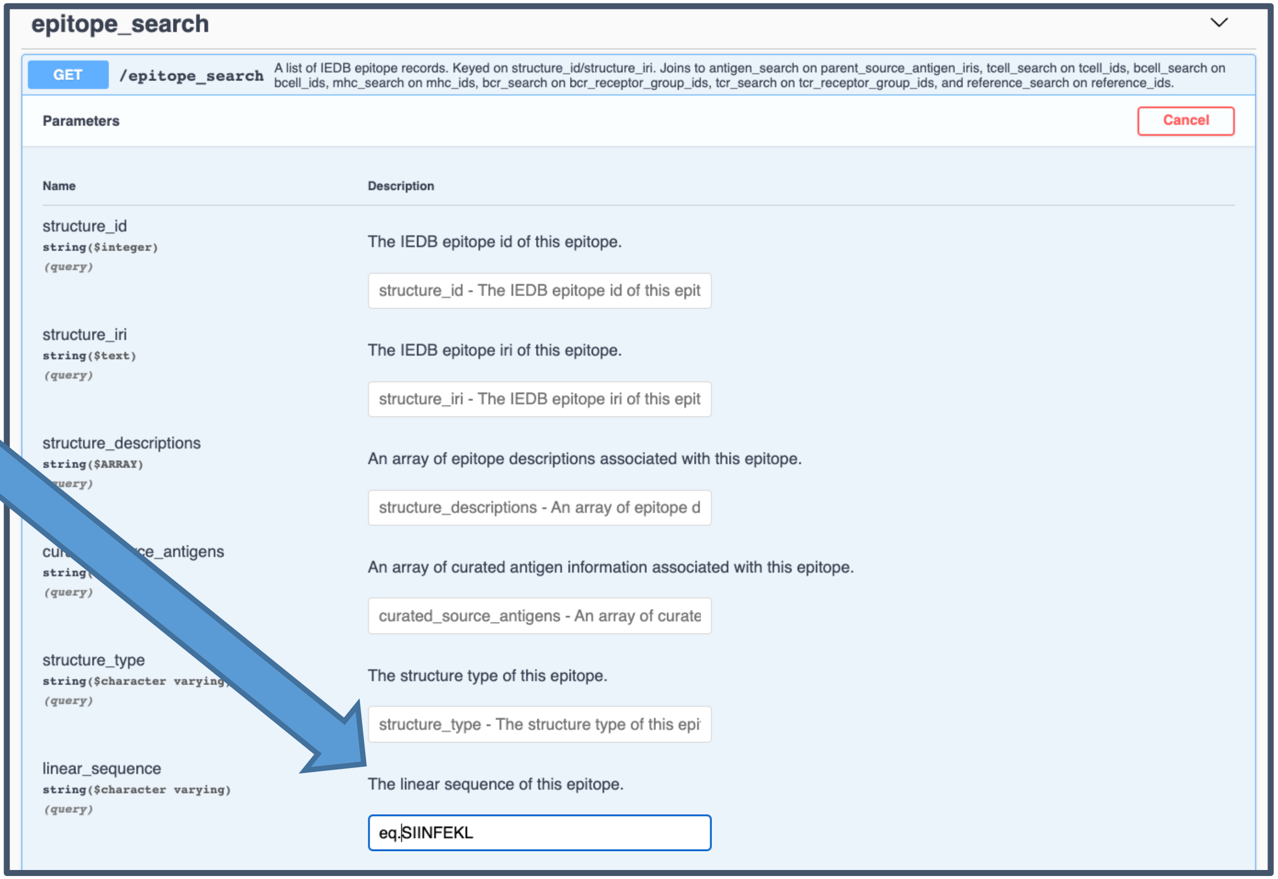
○ Any

● Linear peptide

Exact M: SIINFEKL

○ Discontinuous

○ Non-peptidic



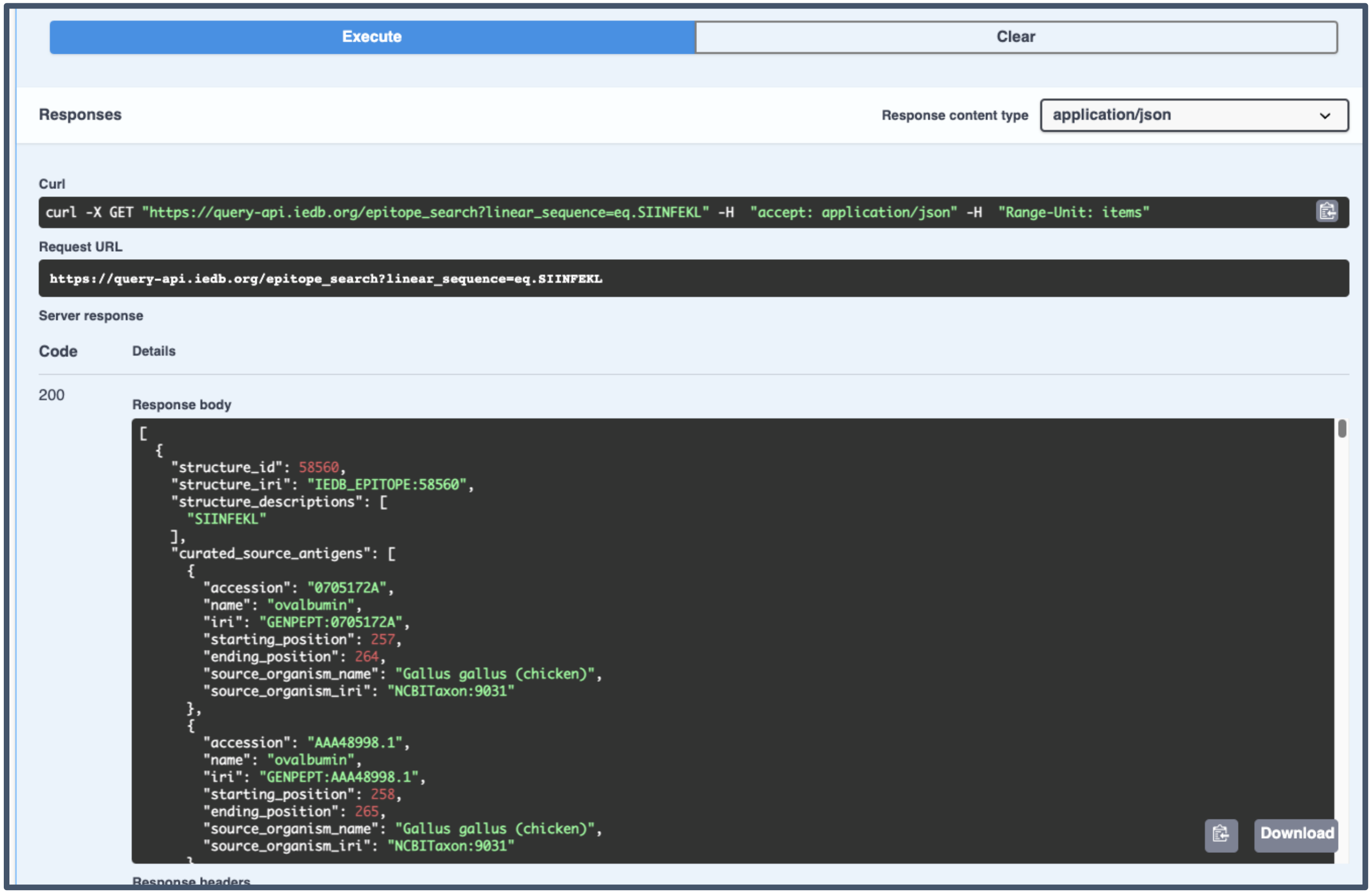
epitope_search

GET `/epitope_search` A list of IEDB epitope records. Keyed on structure_id/structure_iri. Joins to antigen_search on parent_source_antigen_iris, tcell_search on tcell_ids, bcell_search on bcell_ids, mhc_search on mhc_ids, bcr_search on bcr_receptor_group_ids, tcr_search on tcr_receptor_group_ids, and reference_search on reference_ids.

Parameters

Name	Description
structure_id `string($integer)` *(query)*	The IEDB epitope id of this epitope. structure_id - The IEDB epitope id of this epit
structure_iri `string($text)` *(query)*	The IEDB epitope iri of this epitope. structure_iri - The IEDB epitope iri of this epit
structure_descriptions `string($ARRAY)` *(query)*	An array of epitope descriptions associated with this epitope. structure_descriptions - An array of epitope d
curated_source_antigens `string(` *(query)*	An array of curated antigen information associated with this epitope. curated_source_antigens - An array of curate
structure_type `string($character varying)` *(query)*	The structure type of this epitope. structure_type - The structure type of this epi
linear_sequence `string($character varying)` *(query)*	The linear sequence of this epitope. eq.SIINFEKL

Live results returned, along with command-line



Curl

```
curl -X GET "https://query-api.iedb.org/epitope_search?linear_sequence=eq.SIINFEKL" -H  "accept: application/json" -H  "Range-Unit: items"
```

Request URL

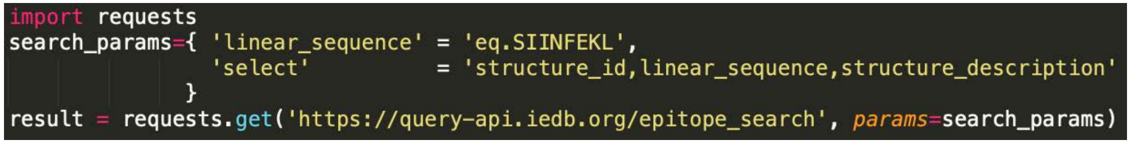
```
https://query-api.iedb.org/epitope_search?linear_sequence=eq.SIINFEKL
```

200

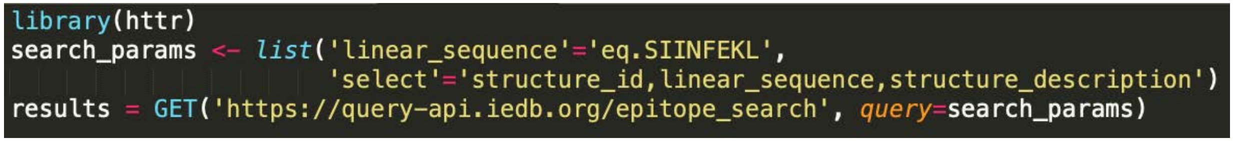
Response body

```
[
  {
    "structure_id": 58560,
    "structure_iri": "IEDB_EPITOPE:58560",
    "structure_descriptions": [
      "SIINFEKL"
    ],
    "curated_source_antigens": [
      {
        "accession": "0705172A",
        "name": "ovalbumin",
        "iri": "GENPEPT:0705172A",
        "starting_position": 257,
        "ending_position": 264,
        "source_organism_name": "Gallus gallus (chicken)",
        "source_organism_iri": "NCBITaxon:9031"
      },
      {
        "accession": "AAA48998.1",
        "name": "ovalbumin",
        "iri": "GENPEPT:AAA48998.1",
        "starting_position": 258,
        "ending_position": 265,
        "source_organism_name": "Gallus gallus (chicken)",
        "source_organism_iri": "NCBITaxon:9031"
```

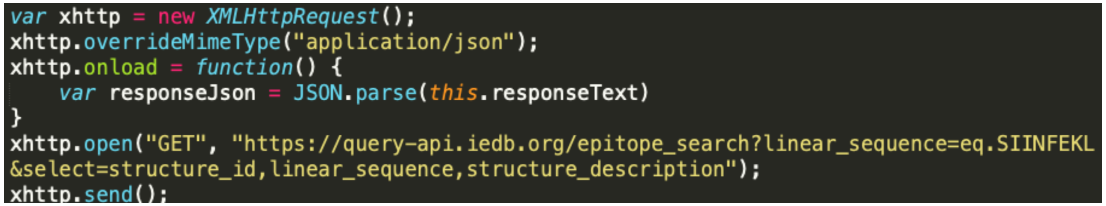
How can I do this in my favorite language?



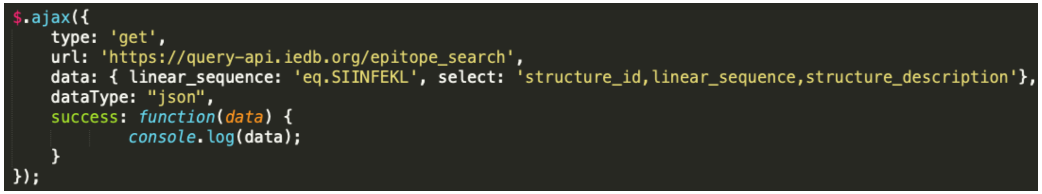
```python
import requests
search_params={ 'linear_sequence' = 'eq.SIINFEKL',
                'select'          = 'structure_id,linear_sequence,structure_description'
              }
result = requests.get('https://query-api.iedb.org/epitope_search', params=search_params)
```



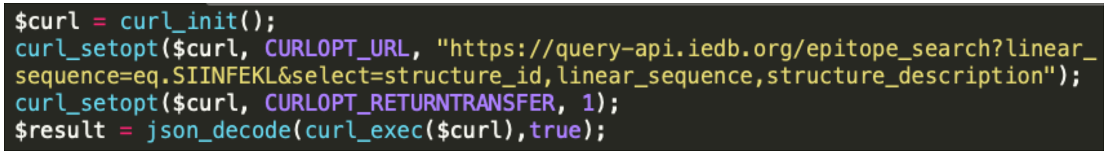
```r
library(httr)
search_params <- list('linear_sequence'='eq.SIINFEKL',
                      'select'='structure_id,linear_sequence,structure_description')
results = GET('https://query-api.iedb.org/epitope_search', query=search_params)
```



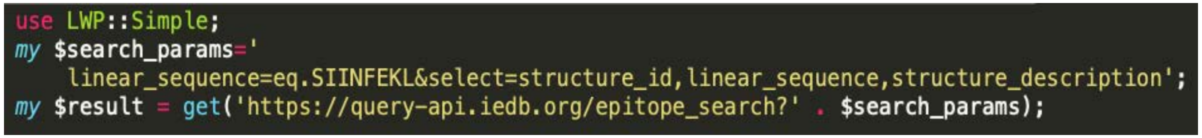
```javascript
var xhttp = new XMLHttpRequest();
xhttp.overrideMimeType("application/json");
xhttp.onload = function() {
    var responseJson = JSON.parse(this.responseText)
}
xhttp.open("GET", "https://query-api.iedb.org/epitope_search?linear_sequence=eq.SIINFEKL
&select=structure_id,linear_sequence,structure_description");
xhttp.send();
```



```javascript
$.ajax({
    type: 'get',
    url: 'https://query-api.iedb.org/epitope_search',
    data: { linear_sequence: 'eq.SIINFEKL', select: 'structure_id,linear_sequence,structure_description'},
    dataType: "json",
    success: function(data) {
            console.log(data);
    }
});
```



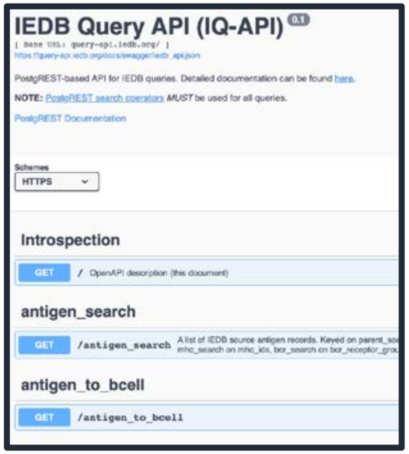
```php
$curl = curl_init();
curl_setopt($curl, CURLOPT_URL, "https://query-api.iedb.org/epitope_search?linear_
sequence=eq.SIINFEKL&select=structure_id,linear_sequence,structure_description");
curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
$result = json_decode(curl_exec($curl),true);
```



```perl
use LWP::Simple;
my $search_params='
    linear_sequence=eq.SIINFEKL&select=structure_id,linear_sequence,structure_description';
my $result = get('https://query-api.iedb.org/epitope_search?' . $search_params);
```

Where can I find more information?

Interactive table & field reference (Swagger)
https://query-api.iedb.org/docs/swagger/



IEDB Query API (IQ-API) 0.1

[Base URL: query-api.iedb.org/]
https://query-api.iedb.org/docs/swagger/iedb_api.json

PostgREST-based API for IEDB queries. Detailed documentation can be found here.

NOTE: PostgREST search operators *MUST* be used for all queries.

PostgREST Documentation

Schemes
HTTPS

Introspection

GET / OpenAPI description (this document)

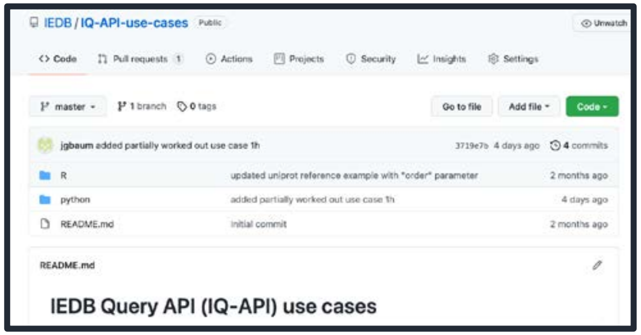
antigen_search

GET /antigen_search A list of IEDB source antigen records. Keyed on parent_so
mhc_search on mhc_ids, bcr_search on bcr_receptor_grou

antigen_to_bcell

GET /antigen_to_bcell

Use cases (Github)
https://github.com/IEDB/IQ-API-use-cases



IEDB / IQ-API-use-cases Public

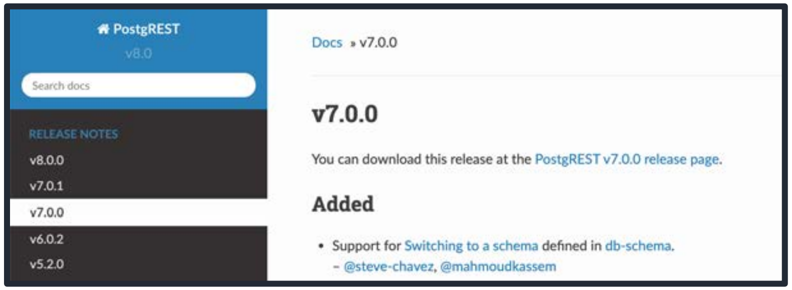
jgbaum added partially worked out use case 1h 3719e7b 4 days ago 4 commits

R	updated uniprot reference example with "order" parameter	2 months ago
python	added partially worked out use case 1h	4 days ago
README.md	initial commit	2 months ago

README.md

IEDB Query API (IQ-API) use cases

PostgREST reference
https://postgrest.org/en/v8.0/releases/v7.0.0.html



PostgREST
v8.0

Docs » v7.0.0

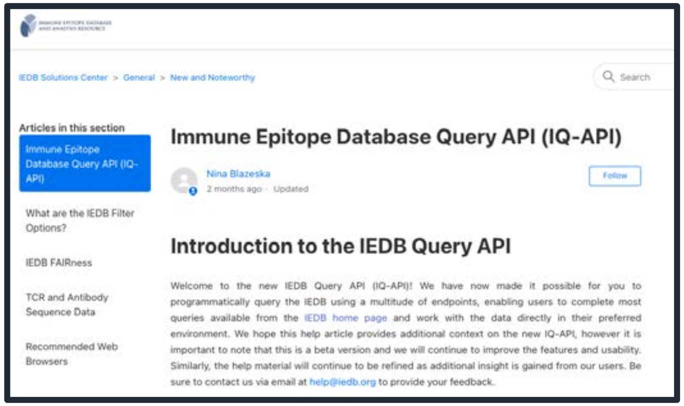
v7.0.0

You can download this release at the PostgREST v7.0.0 release page.

Added

- Support for Switching to a schema defined in db-schema.
  - @steve-chavez, @mahmoudkassem

Overview (Zendesk)
https://help.iedb.org/hc/en-us/articles/4402872882189-Immune-Epitope-Database-Query-API-IQ-API-



IEDB Solutions Center > General > New and Noteworthy
Search
Articles in this section
Immune Epitope Database Query API (IQ-API)
What are the IEDB Filter Options?
IEDB FAIRness
TCR and Antibody Sequence Data
Recommended Web Browsers
Follow

Immune Epitope Database Query API (IQ-API)

Nina Blazeska
2 months ago · Updated

Introduction to the IEDB Query API

Welcome to the new IEDB Query API (IQ-API)! We have now made it possible for you to programmatically query the IEDB using a multitude of endpoints, enabling users to complete most queries available from the IEDB home page and work with the data directly in their preferred environment. We hope this help article provides additional context on the new IQ-API, however it is important to note that this is a beta version and we will continue to improve the features and usability. Similarly, the help material will continue to be refined as additional insight is gained from our users. Be sure to contact us via email at help@iedb.org to provide your feedback.

The IQ-API team

LJI

Nina Blazeska

Michael Talbott

Randi Vita

Bjoern Peters

Jason Greenbaum

Leidos

Kelly Wheeler

Jason Cantrell

Knocean

James Overton